

DESIGN OF AN AIRPORT SURFACE ROUTING EVALUATION TOOL

David J. Martín, Guillermo Frontera, Iñigo Marquínez, Ángel Carrasco, Juan A. Besada
GPDS-CEDITEC, Universidad Politécnica de Madrid, Madrid, Spain

Abstract

There are many studies related with airport surface routing algorithms, based on different approaches and with different evaluation methods and metrics. So, the need of performing a balanced analysis and comparison using a common framework is evident. This paper presents an implementation of an evaluation tool for airport surface routing algorithms. The routing evaluation tool presented here is based in three basic pillars composed by the airport model, the model and generation of traffic and a comprehensive figure of merit function. The paper includes some example evaluations performed over Barajas Airport with representative traffic samples using several simple routing methods.

Keywords: Routing Algorithms, Airport Surface, Evaluation Tool, A-SMGCS.

Introduction

A-SMGCS (Advanced Surface Movement Guidance Control Systems) are designed to meet the needs of safety and efficiency in managing operations at the airport. Route Planning is one of elements of A-SMGCS, related to the optimization of airport surface routes to facilitate the management of the aircraft and vehicles routes on the surface, and maintaining airport performance in high-density situations, even and in low visibility conditions [1].

There are numerous papers that describe algorithms for this problem based on different techniques as linear programming [2-4], graph-based algorithms [5, 6], and combinations of graphs-based algorithms with heuristic search optimization techniques, genetic algorithms [6, 7] or Petri Nets [8]. Each author defines different movement and operational restriction models, in addition to using different airports and traffic samples, so it is not possible to perform a fair comparative analysis of all the different approaches. Furthermore, many academic algorithms include unrealistic assumptions (absence of conflicts, perfect knowledge of aircraft velocities, etc.) making them inapplicable to real scenarios with a real traffic.

Another important aspect observed among the existing algorithms is that they take into account different optimization criteria: some try to minimize distance [5, 7], others optimize total taxi time [2, 3, 4,9], holding time [2, 7], flow capacity of airport [6], etc. As the problem to find an optimal path can have different definitions, so it does not exist a global optimum criterion to decide what is best.

The evaluation methods for the routing algorithms in the existing bibliography are clearly limited. In the cases where evaluation is performed there is no agreement to use common comparative parameters and each author generates particular cases of airports and traffic models to check the performance of the presented algorithms. Several examples of airport models and traffic patterns for the evaluation will be described next.

Reference [2] shows an evaluation example of their algorithm with a small airport model of 28 nodes. To test the taxi-planning in this case, traffic of only 6 taking-off flights and 2 landing aircraft have been simulated, lasting the simulation only 6 minutes. This example seems poor compared to a real case where tens of operations per hour use to be performed. The authors remark that simulations with a full-scale flight schedule with 20 aircraft divided regularly over 15 minutes were performed but the results were not published. To evaluate the algorithm they present the results of the objective function used in the problem based on minimization of total taxi time and total holding time.

In [3] Smeltink et al. perform another simulation with Amsterdam Schiphol airport model and around 20 simultaneous aircraft moving on the surface. The exercise of routing is divided in 12 intervals of 30 minutes corresponding to a congested day. The authors remark the need of testing the algorithm with some other realistic traffic exercises to verify it in a proper way. The performance of three variations of their algorithm is evaluated comparing the mean and the standard deviation of the total taxi time of each case.

Meanwhile, Rathinam et al. in [4] present an evaluation of their algorithm using a portion of the east part of the international airport Dallas-Ft. Worth (DFW) with a set of 25 departing aircrafts which is considered a rush exercise in the mentioned airport. To validate the algorithm they compare with a First-come, first-served (FCFS) ad hoc system. Compared to the FCFS algorithm the authors remark that approximately 6 minutes of taxi time per aircraft are saved, to show this, graphs with the mean taxi times and the average saving time in percentage are depicted.

In [6] Garcia et al. the evaluation of the studied algorithms is performed by means of a simplified directed graph of Madrid-Barajas airport with 24 nodes and 29 arcs. For the first algorithm, that is designed to maximize flow capacity, they use three traffic settings of 6, 13 and 27 departing operations increasing gradually from different terminal buildings in a time interval of 20 minutes. The magnitudes used for the evaluation are flow units per time interval and the time to take-off depending on the number of demanded departures. For the second algorithm presented in this paper, 10 runs have been done to different number of demanded departures that vary between 4 and 18 operations. The magnitudes used for the second case are the time to take-off depending on the number of demanded departures and the flow distribution on the taxiway nodes so comparison with the first algorithm can be done.

Li, Zhen et al. provide in [7] an example based on the Baiyun airport model of Guangzhou considering west runway direction operation. In this example the authors only show the evolution of the objective function used in its algorithm, which includes the conflict waiting time of all aircraft in the network and waiting time for ready aircraft, but do not show other parameters that can be compared with the same algorithm with different traffic or with different algorithms.

In [8] a simple example with a small airport model with only 5 flights is presented. The only comparison performed for this algorithm is made with respect to routes calculated in a static way for the proposed starting and destiny points. The authors remark the importance of the need of simulation with a larger number of aircraft to verify the real benefits of the algorithm.

Gotteland J. et al. in [9] carry out simulations with real traffic of the Roissy Charles De Gaulle airport for a complete day with the three proposed strategies. The compared magnitudes for the evaluation process are the mean delay as function of the number of moving aircraft and the number of aircraft simultaneously moving as function of time for each strategy.

Many other approaches to the problem are perfectly suitable, either combining some aspects of the algorithms described in the previous papers or using different perspectives, different terms to optimize, sets of restrictions or by applying completely alternatives methods to obtain the solution. So a tool that allows comparing algorithms, using a common metric for the evaluation seems necessary. In the following sections we will present an implementation of such a tool.

This paper is organized as follows. In the next section the evaluation tool is presented. Four subsections are included where the airport model, the traffic generation model on the airport surface, the expected outputs for routing algorithms and the composed objective function are explained. In the following section, examples of routing evaluation algorithms are given. Finally in the last section the main conclusions about the evaluation of routing algorithms with the described tool are summarized.

Evaluation Tool System Description

The structure of the routing evaluation is divided in three modules; the first module is the airport model that comprises the data structure to define the airport; the second is the traffic generator which will make use of the information about the apron area and stands, available runways, airport configuration, and some other specific points on the airport layout as runway exits, thresholds, etc. Based on this information the traffic that will be used to evaluate the routing algorithm is generated.

Once the traffic is generated and the routing algorithm processes it, the algorithm will present different output parameters. The main output parameters of a routing algorithm will always be the paths of all aircraft specified by a set of point or nodes on the airport graph and the step time at each node. Depending on the algorithm we want to evaluate, the evaluator algorithm will give different

quality measurements assessing routing results, measuring time-delays, route length (distance) or some other characteristics. These quality measurements will be combined to obtain a global quality metric, usable to quantitatively compare the

routing algorithms with different solutions for different configurations (parameters) of the same algorithm, or different algorithms to each other.

In Figure 1 an algorithm evaluation process is summarized, according to the previous description.

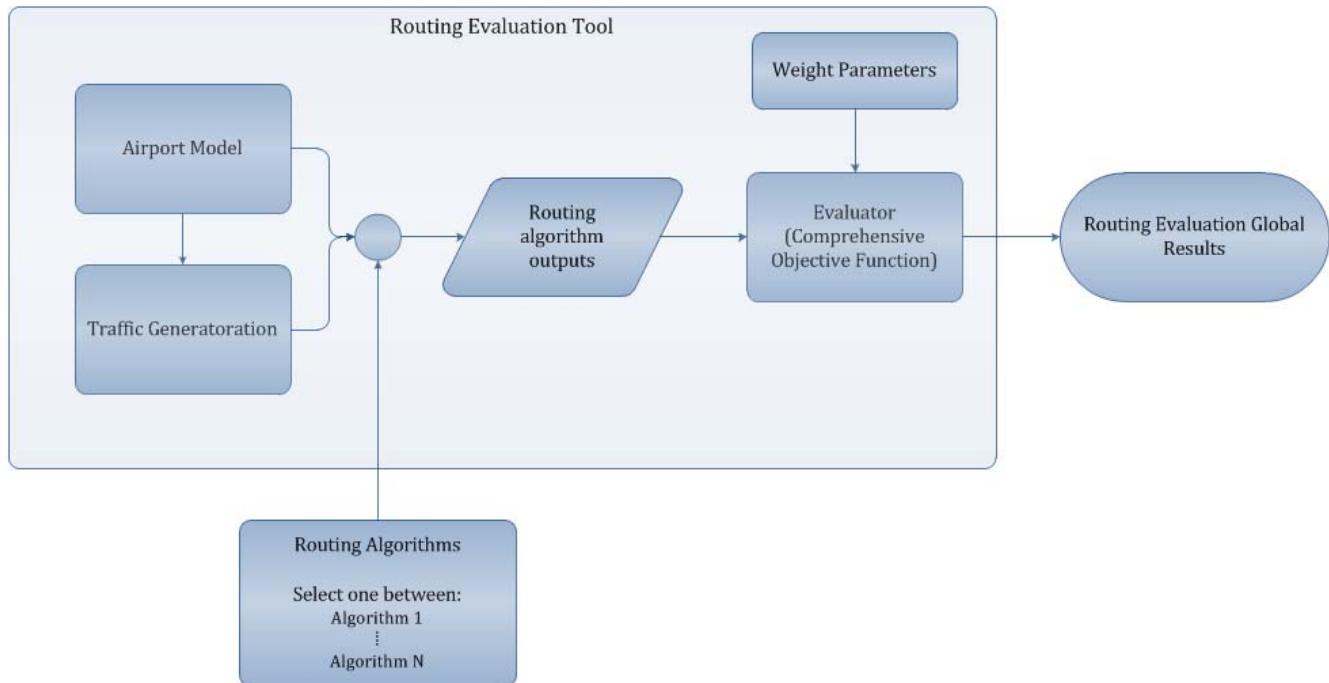


Figure 1. Evaluation Tool Structure

Airport Model

The airport model used for the routing evaluation tool is based on a directed graph (Figure 2) composed of several elements. First of all, the set of nodes of the graph with their positions, corresponding to points with different functions within the airport: airport ramps or stands, intersections, thresholds, runway exits or taxiway junctions. In addition to 2D positions the type of each node (indicated by a label) is part of the model: a node can be designed as a stand or airport ramp, runway threshold, runway exit or regular node.

The nodes will be connected by edges that will indicate the direction to run along the taxiway represented by that directed edge. The edges can then have single or double direction depending on whether the mobiles are allowed to run in one or two

directions on the taxiway segment represented by that edge.

On the airport surfaces the permitted speed on the different taxiways can vary. To represent this matter, each edge connecting the roads of the allowed maneuver areas of the airport model, include a parameter with the information of the velocity allowed when running that edge.

The information about the node's type and the direction of the edges of the graph depend on the applicable airport configuration, directly related to runway operation direction decided by the ATC based on wind direction. If the configuration of the airport changes due to changes of the meteorological conditions or any other mishap, the airport model graph will change to represent the new airport configuration.

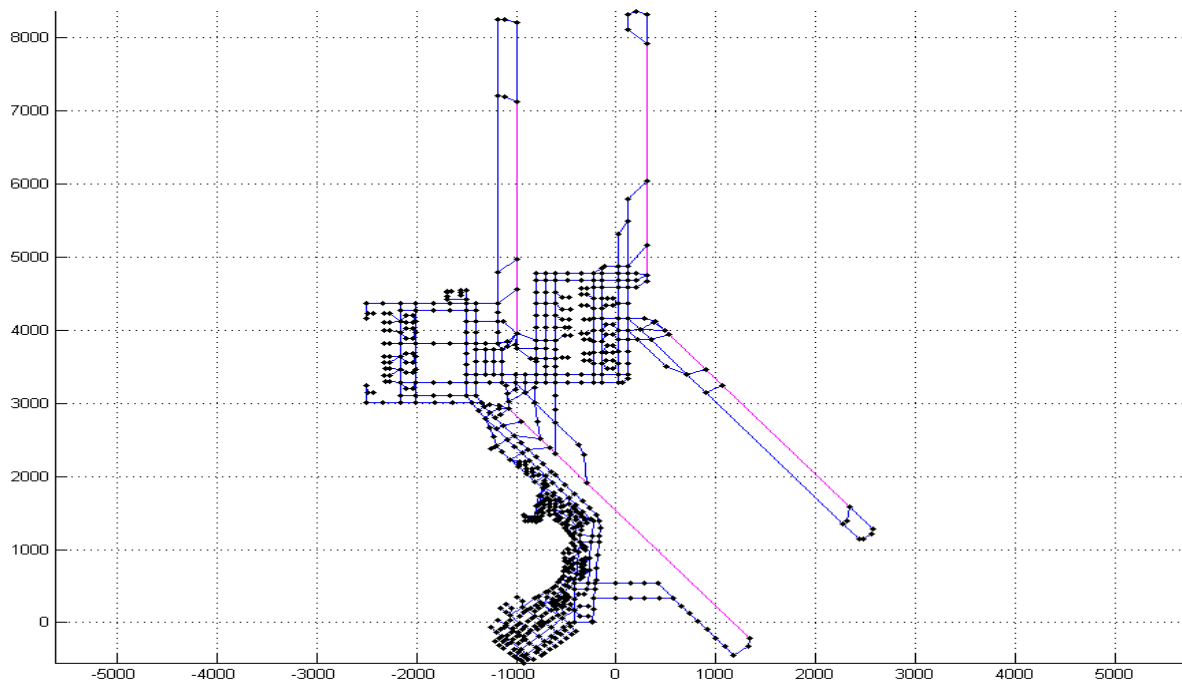


Figure 2. Graph Representation of Madrid-Barajas Airport

Thus the airport model will attempt to picture the airport surface, especially the maneuver area of the airport, to allow the traffic generator and the routing algorithms perform in a proper way.

In Figure 3 a portion of the graph has been depicted overlapped on the real map of the Madrid-Barajas airport to show the level of detail achieved with the graph representation model of the implemented routing evaluation tool. In the picture three out of the four runways of Madrid-Barajas airport and a wide area of taxiways and parking area around the satellite terminal 4 building are shown. All representative nodes are included in this image, since stands, thresholds, runway exits, intersections and crossing nodes are included.

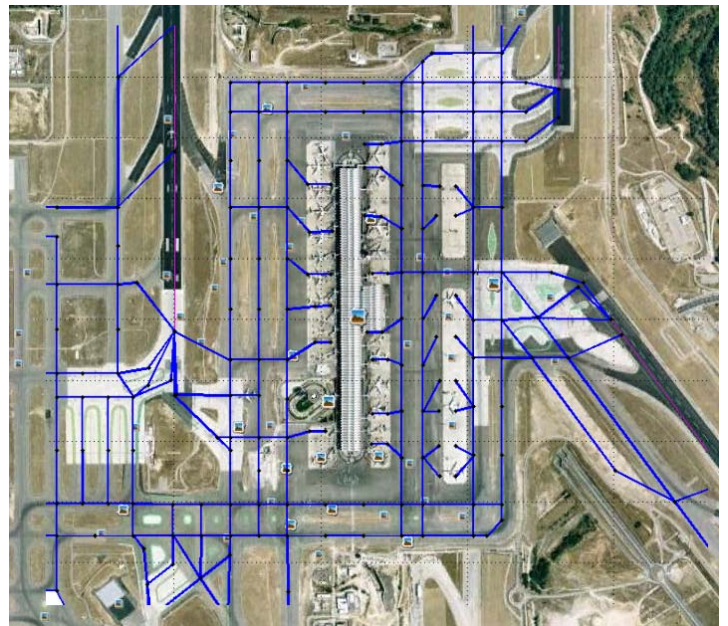


Figure 3. Real Map of Barajas Airport Overlapped with a Graph Representation of Airport Model

The routing evaluation tool can be used for any airport, as one of the inputs of the model is an XML file defining the previous data. The examples included in this paper are based on Madrid-Barajas airport, which was the one available in the defined format at the time of writing of the paper.

In Figure 4 a detail of the directed edges of the graph included in a certain area of Madrid-Barajas airport, for “north” configuration, is depicted. The edges direction can also be shown.

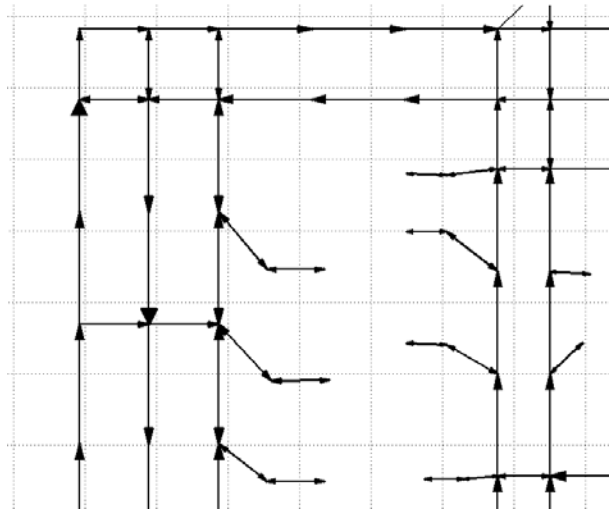


Figure 4. Detail of Edges Indicating Directions for North Configuration of Barajas Airport

Model and Traffic Generation

The second element in the routing evaluation tool is the traffic generator. The routing algorithms must to find, for each aircraft, an optimal path from its gate to a given runway take-off position or from its runway exit to its gate position. So, to evaluate a routing algorithm, traffic based on the real movements performed on surface airport is necessary. This traffic could be based on real movement's recordings or may be synthetic, obtained through simulation means. This second approach enables defining different traffic levels, time and space distributions, and therefore evaluating the algorithms in a broader set of situations. The key aspect for the simulated traffic generation would be its realism: its capability to accurately represent the key aspects of a real traffic. This way the algorithms are evaluated in a realistic way and the information provided by the

routing evaluation tool will be useful to select the more suitable or efficient algorithm.

The inputs of the traffic generation tool are:

- **Airport Model:** the data needed, as input for the traffic generator, are origin and destination nodes types, to tune the inbound and outbound traffic generating individual operations.
- **Configuration file:** Contains all the configuration parameters to adjust the desired simulated traffic, to be detailed below.

The basic traffic generator parameters are duration of the exercise (being possible to generate several hours of traffic or fractions of an hour) and number of operations per hour.

The traffic generator has the capacity to replicate previous exercises by using pseudorandom numbers using a random seed that can be configured by the user in the configuration file.

The traffic generator simulates landing and take-off operations, with the correspondent times at specific points. The landing aircraft do not have a strict time to reach its parking position, so the time generated for these types of flights will be the time instant at the runway exit, however for departing traffic the situation is completely different. Departing aircraft have to fulfill a predefined time corresponding with its take-off slot, then the time assigned to this type of traffic will be the time instant at the take-off scheduled position. The traffic generator has been designed to adjust any temporal gap to get synthetic traffic with landing and taking off traffic that start running almost at the same time or with little mismatch.

For landings an available runway will be chosen and since each runway is provided with different runway exits an exit will be randomly selected. For take-offs an available runway will be selected and the departing traffic generated will have as destiny the take-off position at the assigned runway.

The initial time for each landing aircraft depends on the generated time for the previous aircraft related with the current one by the runway they are operating. They will be separated by a time interval that related to the number of operations per hour. To

give some randomness to that time interval, a uniformly distributed component will be added with a maximum value provided at traffic generator configuration file.

Another characteristic taken into account in the traffic generator is the dependency and independency of operations for parallel runways. For independent runways operations, there will be no constraints to any traffic generated on those runways with respect to the traffic generated in other runways. For the traffic generated on runways with dependent operations with other runways, longitudinal/time separation will be taken into account to generate the traffic on those runways.

Landing aircraft have to be assigned with an available stand or parking area. Initially all stands and ramps are available, once all the traffic is generated the available stands are randomly assigned to the departing aircraft and the rest available stands are assigned randomly again to the incoming flights. The stand where a departing aircraft was parked will be freed (and can be assigned to another aircraft) once the aircraft takes-off. For arriving aircraft, an available stand is assigned in the moment the aircraft takes its runway exit, so since the time instant an aircraft starts running from a runway exit, no other aircraft can occupy the stand assigned to this one. The period of time that the stand is assigned to an aircraft is also configured in the traffic generator, so once a landing aircraft is parked it will have a period of time to occupy its parking position and after that period the stand will be available for the subsequent traffic generation process. This is a limitation of the traffic generator algorithm that should chain take-off and landing operations to the same aircraft.

The output of the traffic generator is a structure with five fields for each operation on airport surface, detailed as follows:

- Type of traffic: Indicates if the generated traffic is an aircraft willing to take off or corresponds to an aircraft that has just landed. The type of traffic is generated as sample of a 50% Bernoulli distribution with two possible values corresponding to departures or arrivals. Thus the number of landings and take-offs will be similar, as usual in airport operations.

- Origin node: The node to be considered the initial point of the path for the routing algorithm. For landing aircraft this node will correspond with a runway exit and for departure flights the origin node is the node located at its parking position.
- Destiny node: This will be the desired node to reach by the current mobile, in case the mobile type is a departing aircraft this node will be a threshold and for landing aircraft will be a node considered as a stand.
- Time at origin/destiny: For landing aircraft will be the starting time of the route at the runway exit point and for departing aircraft will be the time instant at the destiny point or at the moment the aircraft has to reach the take-off position at the correspondent runway threshold.
- Flight Code: Code to identify the specific aircraft is going to perform the route. The traffic generator assigns to each aircraft a decimal code that identifies each aircraft that is moving on the airport uniquely.

Routing Algorithms Outputs

As we have seen before in the papers related with the routing algorithms, there is not a uniform criterion about the evaluation parameters obtained by the different algorithms hence there is not possible a fair comparison between them. To perform the evaluation to any algorithm the first condition is to identify the parameters that can be appropriate and that should be included in the evaluation function. We propose the following list of parameters:

- Distance: Length of each route produced by the algorithm, expressed in meters.
- Taxi time: Is the time difference between the final and the origin time of the operation.
- Number of stops: Minimize the number of stops can be convenient to reduce the number of stopped aircraft within the airport surface and hence reduce the complexity of the operations and increment passenger comfort.
- Waiting time: For algorithms solving conflicts introducing stops along the path,

this will be the accumulated time that an aircraft remains stopped.

- Fuel consumption.
- Direction changes: Paths with many turns are less convenient than others without them, due to reduced comfort and increased operation complexity.
- Slot time displacement: Sometimes, routing algorithms need to perform a displacement of taking-off times within slot limits. These displacements should be minimized to reduce take off time uncertainty.

The defined parameters state different conditions on the efficiency of a particular solution, hence will be necessary to include a normalization of all by means of adequate weighting. The fuel consumption is a parameter difficult to measure and in addition it is related with taxi time and distance. Therefore, it will not be considered in the evaluation tool.

Some of the parameters defined (distance, taxi time) have a minimum value that can be removed in order to make the figure of merit represent actual quality of the routing solution. This can be performed for example with distance and taxi time:

- The minimum distance operation from origin to destination can be calculated without any other traffic affecting the path of the mobile, for instance, by means of the Dijkstra algorithm.
- The minimum time to perform an operation can also be calculated using a shortest path algorithm, but taking into account times to traverse the taxi segments as edge weights, instead of distances.

Therefore, the parameters finally taken into account are:

- The increment in taxi time of a route: $(t^f - t^i) - \Delta t_{ideal}$
- The waiting time along the path: t^w_{Path}
- Length of the route: $\Delta d_{Path} - \Delta d_{ideal}$

- Accumulated turn angles through the path:

$$\alpha_{Path}$$

- Number of stops: S
- The changes in time to the original slot assigned to an aircraft: Δt_{slot}

where:

- t^f : Time at the ending node of the path.
- t^i : Time at the initial node of the path.
- Δt_{ideal} : Minimum elapsed time of an operation.
- Δd_{Path} : Length of the current route path.
- Δd_{ideal} : Length of the ideal route with predefined origin and destiny nodes.

All these parameters have been defined as cost functions; the lower is their value of the parameter the better for the defined route.

Composed Objective Function

We propose composing a comprehensive objective function taking into account all the previous quality parameters. In addition, global performance assessment will somehow average results for all the mobiles on airport surface through the following statistics:

- Average: It is the most typical quality measure. Probability distributions with small average costs can have long tails, meaning there are vehicles with excessive cost associated.
- Maximum value: The disadvantage of this measure is that will be bigger as the size of the sample grows and it will be affected by the presence of outliers. Nevertheless is convenient to know the presence of these outliers, because they can be an indication of problems in the algorithm studied.
- 95th (or 99th) percentile: This measure serves to assess the performance seen by most users. It is therefore the most useful measure for the comparison of different algorithms.

Thus for each output parameter obtained from the evaluation of the algorithm these three statistics will be calculated and the comprehensive objective function will be composed with each of the obtained values properly weighted. Hence we will have the following statistics:

- a_1 = Mean of the time increase of the routes
- a_2 = Maximum of the time increase of the routes
- a_3 = 95th percentile of the time increase of the routes
- b_1 = Mean of the waiting times of the routes
- b_2 = Maximum waiting time of the routes
- b_3 = 95th percentile of the waiting times of the routes
- c_1 = Mean of the distance increase of the routes
- c_2 = Maximum of the distance increase of the routes
- c_3 = 95th percentile of the distance increase of the routes
- d_1 = Mean of the accumulated turn angles along the routes
- d_2 = Maximum accumulated turn angles of the routes
- d_3 = 95th percentile of the accumulated turn angles along the routes
- e_1 = Mean number of stops of the routes
- e_2 = Maximum number of stops of the routes
- e_3 = 95th percentile of the stop's number of the routes
- f_1 = Mean of the changes in slot's time
- f_2 = Maximum change in slot's time
- f_3 = 95th percentile of the changes in the slot's times

Based on the obtained values the comprehensive cost function can be set up with a selection of weights properly chosen. Those weights (P_j) will have the function of adjust conveniently the importance of the statistics in the final quality measure, while normalizing their relative values. The cost function F will be the final obtained value. It is calculated as follows:

$$F = (P_1a_1 + P_2a_2 + P_3a_3) + (P_4b_1 + P_5b_2 + P_6b_3) + L + (P_7c_1 + P_8c_2 + P_9c_3) + (P_{10}d_1 + P_{11}d_2 + P_{12}d_3) + L + (P_{13}e_1 + P_{14}e_2 + P_{15}e_3) + (P_{16}f_1 + P_{17}f_2 + P_{18}f_3) \quad (1)$$

The actual values of the weights should be selected to reflect the operational needs in a certain environment, as the selection of an algorithm against other based on the global function of merit will imply giving more importance to some aspects against others: total taxi time, waiting time, distance, number of conflicts, etc.

In order to select an algorithm or set of parameters it is important the final value of the cost function F , but also the analysis of the individual statistics must be performed, in order to avoid methods, which lead to problematic solutions for lower weight terms. Some solutions to this problem could be changing the F function to a non-linear weighting (i.e. using polynomials), or including constraints over those lower weight terms.

Evaluation Examples

In this section, we present results obtained applying the proposed routing evaluation tool to different routing algorithms. To perform this evaluation, we have used the representation of the Madrid-Barajas airport with a directed graph, composed of 709 nodes including stands, runways exits, runway headers, etc. and 1626 edges which includes bidirectional edges for those nodes connected in two directions. The maximum speed is limited in all the segments to 50 km/h .

The same traffic exercise will be used for all the algorithm's evaluations. The traffic exercise it is one hour long and 120 flights have been generated which it is the maximum estimation of operations in Madrid-Barajas airport.

The airport configuration used is north configuration where runways 36L and 36R are used to take-offs and runways 33L and 33R are used for landing operations.

For the evaluation, we use three different typical routing algorithms, the first one solve the conflicts performing stops for those aircraft which are too close and can cause a safety danger. If there is no possible solution performing stops, then a slot displacement can be performed for the implied aircraft.

The second algorithm solves the conflicts trying to avoid these conflicts by finding a different path, but without performing stops or slot displacements.

While the third algorithm consist of a trade-off between the two previous approaches, deciding for each route if it is a better choice to perform stops or on the contrary it is desirable to avoid edges with possible incoming conflicts.

In Figure 5 and Figure 6 two examples of paths obtained by performing these routing algorithms are shown. The route in Figure 5 corresponds to an incoming flight that lands at the 33R runway and has a destiny stand at the vicinity of terminal 3, while the route in Figure 6 belongs to a departing aircraft that is parked at the surrounding area of terminal 1 and performs it take-off on runway 36L.

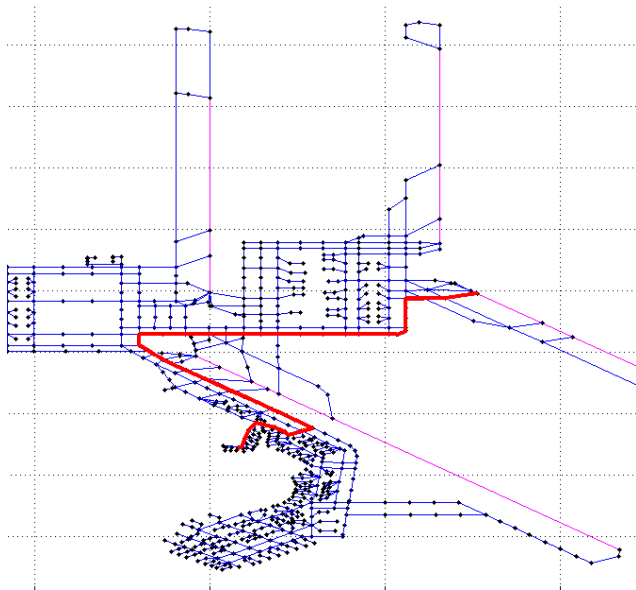


Figure 5. Path Representation Obtained by a Routing Algorithm for an Incoming Aircraft

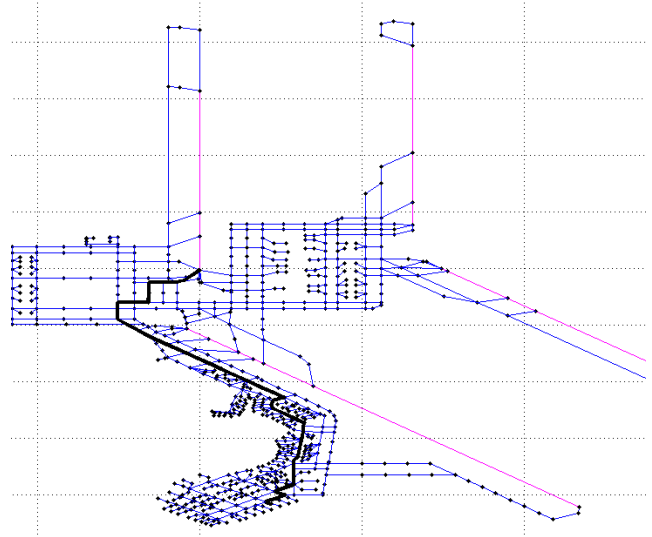


Figure 6. Path Representation Obtained by a Routing Algorithm for a Departing Aircraft

The resulting parameters obtained from the evaluation of each of the previous mentioned algorithms are enclosed in Table 1, Table 2 and Table 3. The units of the parameters included in the tables are seconds for elapsed time, waiting time and time slot displacements, meters for distance and sum of degrees for angles.

Table 1. Evaluation Results of Algorithm 1

	Mean	Maximum	95th Percentile
Elapsed Route Time	17,579	249,09	108,96
Waiting Time	8,25	150	55
Distance	129,57	1915,51	968,02
Angles	5882,52	11552,27	11170,63
Stops Number	0,28	6	1,5
Time Slot Displacement	43,75	480	255

In Table 1 we can observe how the average time for the routes with respect the route without conflicts it is 17,5 seconds, the maximum elapsed route time is 249,09 seconds and that 95% of the routes have an elapsed route time below 108,96 seconds. Comparing with the results in Table 2 we can observe that the elapsed route time for the first algorithm is in general higher that for the second algorithm. As expected, for

the second algorithm that solves the conflicts without stops, the waiting time is zero in all the cases, but we can observe that the route's distance are in general longer than the routes obtained by the first algorithm, that perform the stops. The accumulated angles are slightly higher for the second algorithm due to the larger routes output by this algorithm. Finally the second algorithm does not introduce stops nor time slot displacement, while the maximum number of stops for a route in the first algorithm is 6, the 95% of the routes perform less than 1,5 stops during their paths and the average number of stops for the routes in the first algorithm remains only in 0,28 stops per route. The displacement of the original time slot for the first algorithm has a maximum amount of time of 480 seconds, the 95% of the routes have slot displacement bellow 255 seconds and the average for this parameter is 43, 75 seconds.

Observing the parameters of the Algorithm 1 and Algorithm 2 included in Table 1 and Table 2 respectively we can select the first algorithm in the cases where we are looking to obtain shorter routes with less turns without prioritize the elapsed time of the routes due to the waiting times introduced by the stops. On the contrary the second algorithm will be selected in those cases we prefer larger but more direct routes without produce waits due to stops or slot displacements.

Table 2. Evaluation Results of Algorithm 2

	Mean	Maximum	95th Percentile
Elapsed Route Time	16,77	170,92	96,63
Waiting Time	0	0	0
Distance	233,033	2374	1342,20
Angles	5989,69	11675,26	10690,33
Stops Number	0	0	0
Time Slot Displacement	0	0	0

In Table 3 we can observe the parameters corresponding to the third algorithm where the elapsed times for the routes have been reduced considerably with respect of the two first algorithms. In this case the maximum elapsed time for a route is 63,81 seconds, the average elapsed time will remain in only 7,3 seconds and the 95% of the routes have an

elapsed time bellow 37,07 seconds. The distance and the angles have been also reduced with respect to the previous algorithm result parameters, been this decrease more accused to the distance parameter. The number of stops, waiting time and time slot displacement remains also below the same parameters for the first algorithm.

Table 3. Evaluation Results of Algorithm 3

	Mean	Maximum	95th Percentile
Elapsed Route Time	7,3	63,81	37,07
Waiting Time	1,33	50	5
Distance	82,87	886,27	498,37
Angles	5819,74	11552,27	10491,05
Stops Number	0,05	1	0,5
Time Slot Displacement	32,5	480	255

From this analysis we can conclude that the third algorithm outperform the results obtained by the first algorithm and depending on the desired of having stops and slot displacements we could choose either algorithm 2 or algorithm 3.

The set of weights can be selected to obtain the F function depending on the preference of the algorithm's performance. In this regard, we have selected two different sets of weights to obtain two different values of the F function from the previous three algorithms.

In Table 4 the first set of selected weights is shown, and the resulting F functions to the three different algorithms for the weights in Table 4 are included in Table 5.

Table 4. First Set of Weights Selected

$P_1 = 27,5$	$P_2 = 5,5$	$P_3 = 22$
$P_4 = 45$	$P_5 = 9$	$P_6 = 36$
$P_7 = 0,85$	$P_8 = 0,17$	$P_9 = 0,68$
$P_{10} = 0,55$	$P_{11} = 0,11$	$P_{12} = 0,44$
$P_{13} = 450$	$P_{14} = 90$	$P_{15} = 360$
$P_{16} = 12,5$	$P_{17} = 2,5$	$P_{18} = 10$

With the previous set of weights the minimum value of the F function is obtained by the second algorithm, being the best choice for this case. The algorithm 3 has obtained a value of F higher than the obtained by the algorithm 2 but it clearly improve the resulting value of algorithm 1 (see Table 5).

Table 5. Resulting Values of the F Function for the First Set of Weights

Algorithm Number	F value
1	23969.9
2	14323.8
3	16153.5

The second set of weights is shown in Table 6. In this case the ‘importance’ of the distance has been increased with respect to the previous set of weights and the rest of the values have been reduced in different proportions.

Table 6. Second Set of Weights Selected

$P_1 = 12,5$	$P_2 = 2,5$	$P_3 = 10$
$P_4 = 20$	$P_5 = 4$	$P_6 = 16$
$P_7 = 6$	$P_8 = 1,2$	$P_9 = 4,8$
$P_{10} = 0,4$	$P_{11} = 0,08$	$P_{12} = 0,32$
$P_{13} = 200$	$P_{14} = 40$	$P_{15} = 160$
$P_{16} = 5$	$P_{17} = 1$	$P_{18} = 4$

In Table 7 we can observe the resulting values of the F function for the weights included in Table 6 for the three different algorithms. In this case, the minimum value of the F function is obtained by the third algorithm. The F function of the first algorithm has higher value again but closer to the result obtained by the second algorithm this time.

Table 7. Resulting Values of the F Function for the Second Set of Weights

Algorithm Number	F value
1	20406.1
2	19043.5
3	13282.7

An appropriate adjust of the set’s parameters should be carefully done by the airport operations designer in order to select the most convenient routing algorithm in each case depending on the desired airport’s operational configuration.

Conclusions

With the designed routing evaluation tool has been shown how a fair comparative of the different routing algorithms is possible by means of an airport scenario that it is adjusted to a real airport model and the aircraft traffic moving on it. The set of parameters used in the evaluation process has been chosen from the study of the quality measures of the different existing routing algorithms, so the algorithms can be compared depending on the importance considered by the airport operations designer for the different parameters.

Some limitations and possible future improvements for the traffic generator have been observed and analyzed for future implementations.

First of all, the traffic generator can be improved in some aspects. The stands assignation can be performed in a better way, assigning the stands to the aircraft in a more realistic way, agreed with the airport typical procedures. When an incoming aircraft parks on its assigned stand the same aircraft can be considered then as a future departing flight and the stand will freed when the aircraft takes-off. The distribution of the parking areas has been performed uniformly for all the terminal buildings, but a more realistic scenery would distribute the parking areas depending on the traffic flow supported by each terminal building.

The second future improvement is related to a limitation of the simulated traffic. Current version of the traffic generator only considers and generates landing or departing traffic. It should be extended to consider all other possible situations that can occur in the airport surface area, such as aircraft going to hangar to maintenance, fireman’s vehicles, etc. The main extension needed in this sense is that the traffic generator should generate traffic starting from any node of the airport and ending at any other node.

Additionally we should mention that for proposed cost function F , all the aircraft are considered to have the same cost, independently of the length of their path on the airport surface.

Additional normalized statistics can be defined taking into account with the minimum distance, minimum time, etc. to build an objective function more permissive with those aircraft/vehicles whose path it is longer, because as it is expected, longer path implies more number of conflicts, more delay with respect to the optimum path, etc.

Finally, the routing evaluation tool performs a stationary analysis since its output parameters are calculated for the whole period of time, eliminating transients. An extended analysis taking into account the evolution along time could be used to evaluate routing algorithms in larger periods and obtain information of behavior during peaks of traffic. This could be based on the calculation of the F statistics along time using a sliding window.

References

- [1] http://www.eurocontrol.int/airports/public/standard_page/APR1_Projects_ASMGCS.html
- [2] Roling P.C. and H.G. Visser, 2008 “Optimal Airport Surface Traffic Planning Using Mixed-Integer Linear Programming”, vol. 2008, no., International Journal of Aerospace Engineering, Article ID 732828, 11 pages.
- [3] Smeltink, J.W.; M.J. Soomer, P.R. de Waal; R.D. van der Mei, October 2004, “An Optimisation Model for Airport Taxi Scheduling”, Denver, Colo. USA in Proceedings of the INFORMS Annual Meeting.
- [4] Rathinam, Sivakumar; Justin Montoya; Yoon Jung. 2008, “An Optimization Model for Reducing Aircraft Taxi Times at the Dallas Fort Worth International Airport”. 26th International Congress of the aeronautical sciences 2008.
- [5] Lawson, D.I., 31 Oct-7 Nov 1998, “Concurrent Path Selection and Real-Time Applications: Aircraft Routing and Task Scheduling”, Vol.1, No., Digital Avionics Systems Conference. Proceedings, 17th DASC. The AIAA/IEEE/SAE, pp.C11/1-C11/8.
- [6] Garcia, J.; A. Berlanga; J.M Molina; J.A. Besada; J.R. Casar, 2002, “Planning Techniques for Airport Ground Operations”, vol.1, no., The 21st Digital Avionics Systems Conference, 2002. Proceedings. no., pp. 1D5-1- 1D5-12.
- [7] Li, Zhen; Jun Zhang; Xuejun Zhang, 16-19 Aug. 2009, “A Dynamic Model for Aircraft Route Optimizing in Airport Surface Management”, 9th International Conference on Electronic Measurement & Instruments, 2009. ICEMI '09, pp.3-1068-3-1072.
- [8] Xin-min, Tang; Wang Yu-ting; Han Song-chen, Jan. 2010, “Aircraft Taxi Route Planning for A-SMGCS Based on Discrete Event Dynamic System Modeling”, Vol.1, No., Second International Conference on Computer Modeling and Simulation, 2010. ICCMS '10. pp.224-228, 22-24.
- [9] Gotteland, Jean-Baptiste; Nicolas Durand; Jean-Marc Alliot; Erwan Page, December 2001, “Aircraft Ground Traffic Optimization”, 4th USA/Europe Air Traffic Management Research and Development Seminar, Santa Fe, NM, USA on 04 - 07 December 2001.

Acknowledgements

The authors would like to thank Cristina Bravo, Gema Perez, Helena Puig, Fernando Márquez, Vladimir Coca, Juan Barrio and Ramón Lema from INDRA for the fruitful discussions for this paper, regarding A-SMGCS improvements in the context of UPM-INDRA collaboration for SESAR.

*30th Digital Avionics Systems Conference
October 16-20, 2011*